

In the claims:

Please cancel claims 2, 3, 6, 9, 10, 13, 16, 17, and 20 and amend claims 1, 4, 5, 7, 8, 11, 12, 14, 15, 18, 19 and 21 as indicated below. A complete listing of the claims follows.

1. (Currently amended) A system, comprising:

a storage device configured to store a plurality of files; and
a file system configured to manage access to said storage device, wherein said file system is configured to:

compute a compressed size of at least a portion of a given one of said plurality of files, wherein said given file includes a plurality of ordered portions; and

store an indication of said compressed size in data storage corresponding to said given file, wherein said data storage comprises a named stream;

wherein said file system is further configured to:

store a respective plurality of compression dictionaries in said named stream, wherein each said respective compression dictionary corresponds to one of said ordered portions;

detect a write operation to a given one of said ordered portions of said given file;

in response to detecting said write operation, invalidate the respective compression dictionaries corresponding to said given ordered portion and any higher-ordered portions than said given ordered portion of said given file; and

subsequent to said invalidating, recompute a respective compressed size of only said given ordered portion and any higher-ordered portions than said given ordered portion of said given file.

2.-3. (Canceled)

4. (Currently amended) The system as recited in claim [[3]]1, wherein said at least a portion of said given file corresponds to [[a]]said given ordered portion of said given file, and wherein said file system is further configured to:

suspend computing a compressed size of said given ordered portion of said given file; and

subsequent to said suspending, resume computing said compressed size of said given ordered portion of said given file without recomputing a compressed size of a lower-ordered portion than said given ordered portion of said given file.

5. (Currently amended) The system as recited in claim [[3]]1, wherein a given respective compression dictionary corresponding to a higher-ordered portion of said given file includes a given respective compression dictionary corresponding to a lower-ordered portion.

6. (Canceled)

7. (Currently amended) The system as recited in claim 1, wherein said file system is further configured to:

store a respective compressed size of each of said plurality of files in a corresponding one of a plurality of respective named streams;

compute a compressed size of a concatenated file resulting from appending a first one of said plurality of files to a second one of said plurality of files; and

determine a value of a file harmony metric from said compressed size of said concatenated file and said stored respective compressed sizes of said first file and said second file.

8. (Currently amended) A method, comprising:
storing a plurality of files;

computing a compressed size of at least a portion of a given one of said plurality of files, wherein said given file includes a plurality of ordered portions;
and
storing an indication of said compressed size in data storage corresponding to said given file, wherein said data storage comprises a named stream;
storing a respective plurality of compression dictionaries in said named stream,
wherein each said respective compression dictionary corresponds to one of
said ordered portions;
detecting a write operation to a given one of said ordered portions of said given
file;
in response to detecting said write operation, invalidating the respective
compression dictionaries corresponding to said given ordered portion and
any higher-ordered portions than said given ordered portion of said given
file; and
subsequent to said invalidating, recomputing a respective compressed size of only
said given ordered portion and any higher-ordered portions than said given
ordered portion of said given file.

9.-10. (Canceled)

11. (Currently amended) The method as recited in claim [[10]]8, wherein said at least a portion of said given file corresponds to [[a]]said given ordered portion of said given file, and further comprising:

suspending computing a compressed size of said given ordered portion of said given file; and

subsequent to said suspending, resuming computing said compressed size of said given ordered portion of said given file without recomputing a compressed size of a lower-ordered portion than said given ordered portion of said given file.

12. (Currently amended) The method as recited in claim [[10]]8, wherein a given respective compression dictionary corresponding to a higher-ordered portion of said given file includes a given respective compression dictionary corresponding to a lower-ordered portion.

13. (Canceled)

14. (Currently amended) The method as recited in claim 8, further comprising:
storing a respective compressed size of each of said plurality of files in a
corresponding one of a plurality of respective named streams;
appending a first one of said plurality of files to a second one of said plurality of files to yield a concatenated file;
subsequent to said appending, computing a compressed size of said concatenated file; and
determining a value of a file harmony metric from said compressed size of said concatenated file and said stored respective compressed sizes of said first file and said second file.

15. (Currently amended) A tangible, computer-accessible storage medium comprising program instructions, wherein the program instructions are computer-executable to:

store a plurality of files;
compute a compressed size of at least a portion of a given one of said plurality of files, wherein said given file includes a plurality of ordered portions; and
store an indication of said compressed size in data storage corresponding to said given file, wherein said data storage comprises a named stream;
store a respective plurality of compression dictionaries in said named stream, wherein each said respective compression dictionary corresponds to one of said ordered portions;
detect a write operation to a given one of said ordered portions of said given file;
in response to detecting said write operation, invalidate the respective compression dictionaries corresponding to said given ordered portion and

any higher-ordered portions than said given ordered portion of said given file; and
subsequent to said invalidating, recompute a respective compressed size of only said given ordered portion and any higher-ordered portions than said given ordered portion of said given file.

16.-17. (Canceled)

18. (Currently amended) The computer-accessible storage medium as recited in claim ~~[[17]]~~15, wherein said at least a portion of said given file corresponds to ~~[[a]]~~said given ordered portion of said given file, and wherein the program instructions are further computer-executable to:

suspending computing a compressed size of said given ordered portion of said given file; and

subsequent to said suspending, resuming computing said compressed size of said given ordered portion of said given file without recomputing a compressed size of a lower-ordered portion than said given ordered portion of said given file.

19. (Currently amended) The computer-accessible storage medium as recited in claim ~~[[17]]~~15, wherein a given respective compression dictionary corresponding to a higher-ordered portion of said given file includes a given respective compression dictionary corresponding to a lower-ordered portion.

20. (Canceled)

21. (Currently amended) The computer-accessible storage medium as recited in claim 15, wherein the program instructions are further computer-executable to:

store a respective compressed size of each of said plurality of files in a corresponding one of a plurality of respective named streams;

append a first one of said plurality of files to a second one of said plurality of files
to yield a concatenated file;
subsequent to said appending, compute a compressed size of said concatenated
file; and
determine a value of a file harmony metric from said compressed size of said
concatenated file and said stored respective compressed sizes of said first
file and said second file.